AMENDMENTS TO THE SPECIFICATION:

Please amend the paragraph beginning on page 7, line 24 as follows:

FIG. 4 is a flowchart of a method according to the present invention for downloading a known or requested file wherein file server 204 is untrusted and offers a file to PDA 202 at step 402 subsequent to PDA 202 identifying files or querying file server 204 for download. File server 204 can still be untrusted because PDA 202 is able to verify the checksum of the transferred file via directory server 206. PDA 202 downloads the file at step 404. Once downloaded, PDA 202 provides a description at step 406 of the downloaded file and further provides an indication that file is available for use. PDA 202 then calculates an MD5 checksum at step 408 that is used at step 410 to verify that the checksum and file description are what they purport to be. Directory server ~~412~~ 206 verifies the checksum at step 412. Upon verification, the downloaded file is available for use at step 414. Note that where directory server 206 cannot verify the checksum, the process terminates at step 416 by aborting the download and deleting the file from PDA 202.

Please amend the paragraph beginning on page 11, line 3 as follows:

Shown in FIG. 7 is a flowchart for a method 700 by which PDA 202 continues to download a previously partially downloaded file upon identifying a new access point. PDA 202 can be configured to continually search for available access points such that at step 702, PDA 202 locates an access point. Here, the general case can be assumed where the located access point in method 700 is different from the access point of 600. Accordingly, the associated file server 204 can also be assumed to be different. Because PDA 202 already has part of a desired file, it

requests continuation of such file, for example, MD5ofFile.media, at step 704. Recall that MD5ofFile.media is encrypted with key, K, but file server 204 associated with the present access point does not have such key, K. PDA 202, however, does have such information in the form of the encrypted file MD5ofFile.key. Thus, at step 706, PDA 202 transmits MD5ofFile.key to file server 204. With such transmitted information, file server 204 is then able to recover the key, K, as well as the MD5 checksum using its private key at step 708. File server 204 then confirms that the recovered key, K, actually corresponds to the desired file, MD5ofFile.media, at step 710 by matching the MD5 checksums. If the correspondence is not confirmed method 700 terminates at step 720. If the correspondence is confirmed, file server 204 can then encrypt the requested media file at step 712 and proceed to transmit at step 714 the remainder of the desired file in an encrypted form. PDA 202 then receives the desired file at step ~~714~~ 716.

Please amend the paragraph beginning on page 11, line 21 as follows:

In a continuous manner, PDA 202 detects whether the desired file transfer is interrupted at step ~~716~~ 718. Interruptions in file transfer can occur for many reasons, including loss of wireless connection, loss of power to PDA 202, loss of power to file server 204, memory errors, etc. Where an interruption occurs, method 700 can be reinitiated at step 702. That is, PDA 202 will look for an access point from which it can receive the remaining portion of the desired file. Where no interruption occurs, file transfer continues until the complete file is transferred and method 700 terminates at step 718.

Please amend the paragraph beginning on page 13, line 7 as follows:

Thus, at step 802, PDA 202 transmits MD5(ClientEMF), the MD5 checksum of the encrypted media file (EMF) it, as a client, has downloaded. At this step, PDA 202 also transmits the received MD5ofFile.key. Recall, MD5ofFile.key is derived from the K_{pub}[MD5; K] which can be decrypted by with the corresponding private key, K_{priv}. Accordingly, payment server ~~206~~ 208 uses its private key, K_{Priv}, at step 804 to extract MD5 and K. At step 806, ~~Payment~~ payment server 208 encrypts the media file, MD5ofFile.media, with the obtained key, K, to produce server-calculated encrypted media file, ServerEMF. At step 808, payment server 208 calculates an MD5 checksum of ServerEMF, i.e., MD5(ServerEMF). Recall that at step 802, PDA 202 calculated and transmitted a ClientEMF such that at step 812, these two quantities, ServerEMF and ClientEMF, are compared. If the values are equal, it is confirmed that PDA 202 has downloaded the correct file. Accordingly, at step 816, payment server 208 releases the key, K, that allows PDA 202 to utilize the functionality of the downloaded file. If, however, confirmation fails at step 812, the payment process is aborted at step 814.